AMENDMENTS TO THE SPECIFICATION

Please insert the following into the specification at page 10, line 4:

FIG. 13 shows a diagram of an embodiment of components used in a process for sending an XML document from a sender to a request processor.

FIG. 14 shows a diagram of an embodiment of a process 1400 by which the Request Processor repackages the XML document and send it to the recipient.

FIG. 15 shows a diagram of an embodiment of a process 15XX, which packages the recipient's return values and transmits them back to the sender.

Please insert the following excerpt from Appendix 1, pages 22 and 23 into the specification at page 27, line 5:

Referring to FIG. 13, a diagram of an embodiment of a process 1300 by which a Sender 1302 (shown as System B Customer Class) creates an XML document and sends it to a Recipient 1402 (FIG. 14) is shown. The Sender 1302 uses a MessagingController factory class at 1306 to create a new MethodCall object at 1308. The MethodCall object identifies the method to be invoked at the Recipient 1402 and includes the parameters the Sender 1302 wishes to pass to the method. The Sender 1302 uses MessagingController at 1310 to create an Encoder object that is sent to an Encoder at 1312. The Encoder object identifies the type of encoding required by the Recipient 1402. The Sender 1302 uses the Messaging Controller at 1314 to create a new RuntimeRequest object at 1314. The MethodCall and Encoder objects are passed to the new RuntimeRequest at 1316. The Sender 1302 tells the RuntimeRequest object to run a FireURL method at 1318. At this point, the RuntimeRequest object tells the Encoder object to create a message containing the method and parameter information contained in the MethodCall object. This message is encoded in the manner required by the Recipient 1402, which is specified by the Encoder object. Since SOAP is the default Messaging Platform encoding mechanism, all encoded XML documents are considered SOAP-encoded.

The XML document is then attached to a URL, which the RuntimeRequest object sends to the Request Processsor at 1320. The URL created in this step contains information that matches part of the URL sent to the Request Processor by the Recipient 1402 during a registration process. The information that the Recipient 1402 sent in the URL at registration is saved in the Recipient's unique RuntimeRequestHandler, which is stored by the Request Processor. The URL includes the following information: http://server/mode-id/encoding-id/url + XML document.

Please insert the following excerpt from Appendix 1, page 27 into the specification at page 27, after the paragraph above:

When Sender 1302 sends information to a Recipient 1402 that resides on a different system, a connection information file is created for the Recipient1402 that defines:

a host on which the Recipient 1402 resides,

a port on which the Recipient 1402 resides,

a mode-ID,

an encoding ID.

The name of the connection information file becomes synonymous with the name of the connection.

Please insert the following excerpt from Appendix 1, page 23-24 into the specification at page 27, after the paragraph above:

Referring to FIG. 14, a diagram of an embodiment of a process 1400 by which the Request Processor repackages the XML document and sends it to the Recipient 1402 is shown. The Request Processor receives the URL from the Sender 1302. The Request Processor 1304 identifies the name of the RuntimeRequestHandler to be used for this request at 1404. The name of the RuntimeRequestHandler corresponds to the URL-ID contained in the URL.

At 1406, the Request Processor determines the type of Content Handler that must be created for the request based on an Encoding ID included in the URL. The

Request Processor then makes a new ContentHandler object for the specified Encoding ID at 1408.

When the XML document is received by the Request Processor from the network at 1410, the document is no more than a stream of bytes. This stream is converted into an object that the Messaging Platform can use. To do this, the Request Processor tells the Content Handler to run a getContent method at 1412, which converts the stream of bytes into the appropriate object. For example, if SOAP is the encoding type, the appropriate object is a SOAP-encoded XML document.

The Request Processor tells the RuntimeRequestHandler to get an Invokeable at 1414 and then sends the data contained in the XML document created at 1408 to the Invokeable at 1420. The Request Processor tells the Invokeable to run an Invoke method, which runs the method with the parameters specified in the data contained in the XML document.

Please insert the following excerpt from Appendix 1, page 25 into the specification at page 27, after the paragraph above:

Referring to FIG. 15, a diagram of an embodiment of a process 1500 is shown that packages the Recipient's return values and transmits them back to the Sender. The invoked method may return a value, which is sent to the Request Processor at 1502. The Request Processor sends the return value to the ContentHandler at 1504. The ContentHandler was created when the original message was sent (see at 1408 FIG. 14). The ContentHandler contains a method called forResult, which takes the return value and returns it in the particular manner required by the Sender 1302, for example, SOAP encoded. The ContentHandler sends the returned object to the Request Processor at 1506. The Request Processor sends the return value to the Sender 1508.